IN THE SPECIFICATION:

Please amend the specification as follows:

Please amend Paragraph 16 on Page 3 as follows:

[16] An embodiment of the present invention may introduce a validation FUB 150 into an agent 100. The validation FUB 150 may be coupled to the ~~EBC 160~~ EBC 140 to observe transactions posted on the external bus 130. In response to a predetermined triggering condition, the validation FUB 150 may generate data for a new transaction, called a "harassing" bus transaction, to be posted on the external bus 130. Several different types of triggering conditions are possible. When a new external bus transaction is observed, a harassing transaction may be generated if a request type of the new transaction matches a predetermined type. The first external bus transaction is said to be a "triggering" transaction. Alternatively, all external bus transaction (except harassing transactions from the validation FUB 150) may be triggering transactions. In this case, harassing bus transactions would be generated for every transaction on the external bus 130.

Please amend Paragraph 41 on Page 10 as follows:

[41] The cache 450 may be an internal memory. As is known, relative to core cache memories (not shown), the cache 450 typically possesses much greater capacity. For example, a typical cache 450 may be a ~~256 memory~~ 256K memory. By contrast a core data cache may be a 16K memory and a core instruction cache may be an 16K memory. The cache 450 may be a unified cache, one that stores both instruction data and variable data (collectively, "data"). The BSQ 400 also may interface with higher levels of cache (not show), which may 3H or more in size.

Please amend Paragraph 42 on Page 11 as follows:

[42] Read requests from the arbiter 440 may be input to both the cache 450 and to the internal queue 460. The cache 450 may include control logic (not shown) that can determine whether the requested data is stored in the cache 450. If so, the request is said to "hit" the cache 450. The cache 450 will furnish the requested data to the core 420 over a communication path (also not shown). Otherwise, the request is said to "miss" the cache. The cache 450 may communicate a hit or a miss to the internal ~~queue 430~~ queue 460 over a line 452.

Please amend Paragraph 43 on Page 11 as follows;

[43] The internal queue 460 may include control circuitry and buffer memory to process requests from the ~~arbiter 430~~arbiter 440. The internal queue 460 also receives hit/miss indicators from the cache 450. If a read request hits the cache 450, the internal queue 460 may permit the queued request to terminate as it advances out of the queue 460. But if a read request misses the cache 450, the request should be completed by retrieving the requested data from an external memory (not shown). In this case, when the read request advances out of the internal queue 460, the internal queue 460 may cause the request to be entered in the external transaction queue 470.

Please amend Paragraph 50 on Page 12 as follows:

[50] As noted, the validation FUB may be provided within an agent in a computer system. A validation FUB need be provided in only one of the agents in the system to be able to stress test the system. FIG. 6 illustrates an exemplary computer system 500 according to an embodiment of the present invention. The computer system 500 may include multiple agents ~~510-560~~ 510, 520, 530, 540, 550, and 560, each coupled to a common communication bus 570. Of the agents, four are shown as processors ~~510-540~~ 510, 520, 530, and 540. Other agents include a system memory 550 and an IO interface 560. A validation FUB 515 is illustrated as being a member of one of the processors 510 but, alternatively, could be provided in one or more of the other agents ~~520-560~~ 520, 530, 540, 550, and 560.

Please amend Paragraph 51 on Page 13 as follows:

[51] FIG. 7 illustrates another exemplary computer system 600 according to an embodiment of the present invention. Again, multiple agents ~~610-630~~ 610, 620, and 630 are coupled to a common communication bus 640. In this example, only one agent 610 is shown as a processor. A memory controller 620 and IO interface 630 also are shown in FIG. 7. In this example, a validation FUB 625 is shown as a member of the memory controller. Alternatively, the validation FUB could be a member of the IO interface 630 (not shown).